

Appendices

Spatiotemporal Non-Uniformity-Aware Online Task Scheduling in Collaborative Edge Computing for Industrial Internet of Things

APPENDIX A PROOF OF THEOREM 1

A.1 The optimal task scheduling decision forms a directed acyclic graph.

Proof. We demonstrate this through proof by contradiction. Assume that a cycle exists in the optimal task scheduling decision graph, as illustrated in Fig. 1(a). Without loss of generality, we assume that $a > b > c$. Consequently, Fig. 1(a) and Fig. 1(b) are equivalent regarding the total computational latency and energy consumption. However, regarding task request scheduling latency and cost, it is evident that Fig. 1(b) outperforms Fig. 1(a). Therefore, Fig. 1(a) does not represent the optimal task scheduling decision graph; thus, the hypothesis is invalid, and the proof is complete. ■

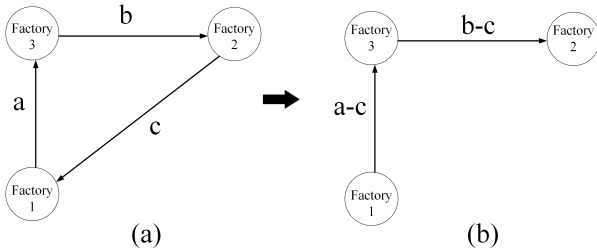


Fig. 1. An example of the task scheduling decision graph with a cycle.

A.2 The optimal task scheduling decision graph does not contain intermediate nodes.

Proof. Similar to the previous proof, assume there is an intermediate node in the optimal task scheduling decision graph, as illustrated in Fig. 2(a), with the assumption that $a > b$. Consequently, Fig. 2(a) and Fig. 2(b) are equivalent in terms of computational latency and energy consumption. However, regarding task scheduling latency and energy consumption, it is evident that Fig. 2(b) performs no worse than Fig. 2(a), as determined by the routing algorithm. Therefore, Fig. 2(a) is not the optimal task scheduling decision, meaning the hypothesis is invalid, and the proof is complete. ■

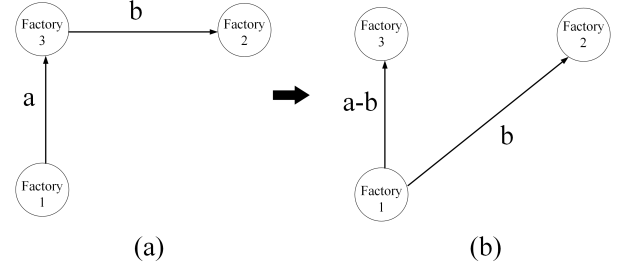


Fig. 2. An example of the task scheduling decision graph with an intermediate node.

A.3 The node in set $\{m | F_m^k(t) = 0\}$ must be a source node.

Proof. When ES m has not allocated resources for the k th service, all requests for the k th service initiated by IIoT devices in factory m must be dispatched to other ESs that have allocated resources for the k th service for processing. Clearly, ES m will not receive requests for the k th service from other ESs. Thus, ES m is a source node with only outgoing edges (out-degree) and no incoming edges (in-degree). ■

A.4 For any ES m corresponding to a source node, if $F_m^k(t) = 0$, the sum of the weights of all outgoing edges equals $N_m^k(t)$. If $F_m^k(t) > 0$, the sum is less than or equal to $N_m^k(t)$.

Proof. When $F_m^k(t) = 0$, ES m has not allocated resources for the k th service. Thus, all requests for the k th service from IIoT devices in factory m must be dispatched to other ESs that have allocated resources for the k th service for processing. Consequently, the sum of the weights of all outgoing edges equals $N_m^k(t)$. Similarly, if $F_m^k(t) > 0$, the sum of the weights of all outgoing edges from this vertex must be less than or equal to $N_m^k(t)$. ■

A.5 A sink node must exist whenever there is a source node.

Proof. From the previous proofs, we conclude that the optimal task scheduling decision can be represented as a directed acyclic graph without intermediate nodes. Therefore, there must also be a sink node whenever a source node exists. ■

APPENDIX B

PROOF OF THEOREM 2

Based on equation (15) in the main manuscript, we can derive

$$Q_k(t+1) \geq Q_k(t) + E_k(t) - E_k^{avg}. \quad (1)$$

Shifting the terms results in $E_k(t) - E_k^{avg} \leq Q_k(t+1) - Q_k(t)$. By summing over all time slots, we can obtain

$$\sum_{t=0}^{t=T-1} (E_k(t) - E_k^{avg}) \leq Q_k(T) - Q_k(0) = Q_k(T). \quad (2)$$

Finally, the time-averaged value is obtained by taking the average of both sides over time:

$$\sum_{t=0}^{t=T-1} \frac{1}{T} (E_k(t) - E_k^{avg}) \leq \frac{1}{T} Q_k(T). \quad (3)$$

Thus, when the virtual queue is stable (i.e., the value of $Q_k(T)$ has an upper bound), the right-hand side of equation (3) approaches zero as $T \rightarrow \infty$, thereby satisfying the long-term constraint.

APPENDIX C

PROOF OF THEOREM 3

Based on equation (15) in the main manuscript, we can derive

$$Q_k(t+1)^2 \leq (Q_k(t) + E_k(t) - E_k^{avg})^2, \quad (4)$$

i.e., $Q_k(t+1)^2 \leq Q_k(t)^2 + (E_k(t) - E_k^{avg})^2 + 2Q_k(t)(E_k(t) - E_k^{avg})$.

By shifting the terms and dividing both sides of the inequality by two, we can obtain

$$\begin{aligned} \Delta(Q_k(t)) &= L(Q_k(t+1)) - L(Q_k(t)) \\ &= \frac{1}{2} Q_k(t+1)^2 - \frac{1}{2} Q_k(t)^2 \\ &\leq \frac{1}{2} (E_k(t) - E_k^{avg})^2 + Q_k(t)(E_k(t) - E_k^{avg}) \\ &\leq B + Q_k(t)(E_k(t) - E_k^{avg}). \end{aligned} \quad (5)$$

APPENDIX D

PROOF OF THEOREM 4

To analyze the computational complexity of problem \mathcal{P}_2 , we first examine its optimization variables and objective function. Based on constraint (14b) in the main manuscript, problem \mathcal{P}_2 is classified as an integer programming problem. As it is constructed using the Lyapunov drift-plus-penalty framework, the objective function incorporates a penalty term. Moreover, as shown in equations (3)-(12), the objective function contains fractional terms. Hence, \mathcal{P}_2 can be categorized as an integer optimization problem involving penalized fractional terms. According to the proof in [42], such problems are NP-hard in general. In addition, as shown in equation (15), the objective function of \mathcal{P}_2 includes a piecewise nonlinear term, whose non-smooth structure further increases the problem's complexity. Therefore, problem \mathcal{P}_2 is NP-hard.